METHOD FOR ENSURING THE INTEGRITY OF A DATA RECORD SET

FIELD OF THE INVENTION

[0001] The invention relates to a method, system and computer program for ensuring the integrity of data record set stored on a database or a similar information storage.

BACKGROUND OF THE INVENTION

[0002] Many computerized applications produce huge amounts of data to be stored. Typically events of the computerized applications are logged into a log file. The log files are one of the most important sources of information for system operators, software developers, security personnel and various other groups.

[0003] Traditionally log data files are written in a sequential manner into the log file. The basic elements of most types of the log files are log records that are often represented as rows in a log file. It is very important that the structure and contents of a log file remain authentic. Especially for security monitoring it is important that the rows may not be modified or deleted in any way without administrator noticing made changes.

[0004] Well-known methods for ensuring the integrity of a log file exist already today. For example, message authentication codes (MAC) or digital signatures can be used to associate a cryptographical code with each log file. Later unauthorized modifications can be detected because the digital signature or authentication code changes, if the contents of the file change. However, these kinds of methods do not protect

the integrity before the digital signature or another kind of authentication code is assigned to the file to be protected.

[0005] However, in many applications the amount of data needed to be stored is huge. Thus, there is a need for storing log data or similar data to a relational database. There the question of integrity protection is somewhat different. In relational databases data is stored in tables consisting of tuples of attributes, so called records. Typically log entries are stored on a database so that each log row corresponds to a record of a particular database table.

[0006] Integrity protection in relational databases relies traditionally on restricting the access rights of the users of the database so that unauthorized users may not alter the contents of the database. Access control is enforced by the relational database management system (RDBMS). Another way of ensuring the integrity of a database is to save it to a disk file and to attach a cryptographic code to it as described above.

[0007] This approach is often impractical as many database tables are dynamic by their nature and have to be updated very often. In a log database, for instance, log entries generated during a day have to be inserted into the corresponding database table all the time as the amount of the data to be stored may be huge, as in bank transactions. Freezing the database table's contents and protecting its integrity with a cryptographic checksum is only then useful, when one can be sure that the contents of a table will not have to be updated anymore. In a log database this means

that one has to use per-day database tables for storing the information. One drawback of such a solution is that queries, which access several days' data, have to make several table lookups to execute a query.

[0008] US 5978475 (Schneier et al.) discloses a method for verifying the integrity of a log file. However, the aforementioned patent does not disclose any means for arranging the data on a database in which the administrator has full capabilities to modify the data in data records.

[0009] A major deficiency of traditional solutions is also that they cannot be applied in a setting, where a database system is used and the database administrator cannot be entirely trusted. In most RDBM systems the database administrator (DBA) has close to unlimited authorizations to modify the database and its contents. Any data that is inserted into the database may be modified by a malicious administrator even before the data is cryptographically protected from unauthorized modifications.

[0010] A major drawback of the prior art is the problem of controlling access rights to the database. A further drawback is that the data cannot be stored on files to be digitally signed as the files change all the time. A third major drawback is that the database administrator must be trusted. Nowadays the administrator is typically a technician who actually would not even need to know the information stored on a database. Thus, there is a need for a method, which allows a plurality of people to view and check the integrity of the contents of a database while having access rights for storing data to the database.

SUMMARY OF THE INVENTION

[0011] The invention discloses a method for ensuring data integrity in database systems. The invention discloses a solution for having publicly viewable databases with publicly available integrity checksums that can be used for integrity verification. According to the present invention the integrity checksum is computed with a cryptographic method from the data to be stored, a checksum of the previous record and a storage key. The storage key is issued only to entities that have a permission to sign the data on the database. A signing entity may and should be different from the database administrator. One solution is to use public key cryptography in which the signing entity calculates an integrity checksum with his/her private key and people willing to verify the integrity may use his/her public key for verification. The calculated integrity checksum is then attached to the data record. The first record may be a generated initial record or it may harness a previously agreed previous checksum that is needed to compute its own checksum. In the verification the integrity checksum is computed similarly and compared to the previously computed checksum attached to the specific data record.

[0012] The benefit of the invention is to allow an authentic database with integrity checks. With the method according to the invention the database can be signed so that only the signing authority may change the contents of the database. According to the invention data records stored on a database may not be de-

leted or altered in any way without breaking the chain of computed integrity checksums.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The accompanying drawings, which are included to provide a further understanding of the invention and constitute a part of this specification, illustrate embodiments of the invention and together with the description help to explain the principles of the invention. In the drawings:

[0014] **Fig. 1** is a flow chart illustrating the basic principle of integrity verification according to the invention,

[0015] **Fig. 2** is a flow chart illustrating one embodiment of storing a data record according to the invention,

[0016] **Fig. 3** is a block diagram illustrating an embodiment of the system according to presented in Figure 2.

DETAILED DESCRIPTION OF THE INVENTION

[0017] Reference will now be made in detail to the embodiments of the present invention, examples of which are illustrated in the accompanying drawings.

[0018] Figure 1 discloses a flow chart illustrating the basic principle of integrity verification. According to Figure 1 input data can be received in any suitable form. However, the invention is most useful in cases in which there are a lot of data entries arriving at a fast pace. Suitable entries can be for ex-

ample data records of the log files of bank transactions that are typically stored in large databases. These log files must be authentic and they must include every event so that they would be accepted in the court of law if necessary.

[0019] According to Figure 1 data arrives to a signing entity 10. Signing entity 10 has its own administrator with authorization to sign data records. Signing may be in the form of digital signature, encryption, or one-way hash. In this description, signing refers to the process of computing a checksum and attaching the computed checksum to the data record. Later on a signing key is referred to as a storage key that may be any type of signing key. On the other hand, it might be useful to use traditional public key encryption method to allow including the name of the signatory to each signed record. The key may be inserted to the system similarly as in secure mailing systems in which the key comprises a secret key file and a secret password part that is typed to the encryption device. The key may also be inserted with a smart card or similar or with any other suitable device.

[0020] The method according to the invention signs each data record with an integrity checksum that is computed from the data record to be signed, an integrity checksum of the previous record and the storage key. The computed integrity checksum is then attached to the data record. It may be attached to the data itself or a database 11 may contain a separate field for the integrity checksum. As the computed integrity checksum depends on the previous integrity checksum, it is not possible to remove one or more lines from

the middle of the records without breaking the integrity, as the complete chain of integrity checksums is needed for verification. Signed data with integrity checksums will be stored on database 11. Database administrator may perform various tasks to stored data, but he/she cannot change the contents of the data nor remove data records secretly.

[0021] The verification of the integrity of consequent data records is performed similarly as signing. A verification entity 12 computes an integrity checksum based on the data record to be signed, a previous integrity checksum and storage key. The computed integrity checksum is then compared to the checksum stored on database 11. If the checksums are not equal, the database has been changed and it is not authentic. The method is beneficial as the integrity of a data record can be checked rapidly without a need to check the integrity of whole database. Verification can be started at any point in the stream of consecutive data records. It should be noted, that the authenticity of the record from which the previous integrity checksum is retrieved cannot be guaranteed. Thus, the verification process must be initiated by retrieving the integrity checksum of the data record previous to the data record to be verified.

[0022] If public key cryptography is used for signing, the signing authority signs records in signing entity 10 with his/her private key. The key may be created for signing for a specific database and may be shared with a trusted group having an authorization for signing. In the verification of the integrity the public key of the signing authority is used for decrypting the checksum.

[0023] There are different ways for starting the database. An initialization vector may be used instead of a previous integrity checksum for the first row of the database, as there is no previous integrity checksum available. The first row may include actual data or data related to the initialization. For example, an initialization vector may comprise information relating to the initialization, such as date, and the digital signature of a responsible person as a checksum. Thus, there is a previous checksum for the first real data record. The initialization vector or row may be applied also in the middle of the database to allow arranging the data into blocks. Arranging data into blocks does not change the verification procedure.

[0024] Figure 2 illustrates a flow chart of one embodiment of storing a data record. At step 20 the data is received from any suitable information system. The data is similar as in embodiment according to Figure 1. After receiving the data an integrity checksum is computed at step 21. The integrity checksum may be computed with a desired commonly known method as disclosed in the embodiment according to the Figure 1. The integrity checksum is computed based on the previous checksum, which refers to the checksum attached to the previous data record, the data to be signed and the storage key. Only persons having authorization to sign data records know the storage key. Previous checksum is read from the memory of the signing device. If the integrity checksum is always read from a database, a malicious database administrator may delete the last row of the database without any problems, as the chain of the integrity checksums will not break. There is also other means for ensuring the authenticity of the last row, for example having a run-

ning sequence number as a part of the checksum parameters.

[0025] The data record is signed by attaching the computed integrity checksum to the data record as illustrated at step 22. The signed data will be stored on the database. The database may contain separate fields for the data and the integrity checksum. The database may also contain additional information fields that may also be used for computing the integrity checksum, for example name of the signatory. After storing the data on the database, the integrity checksum is stored on a memory of the signing device, as illustrated at step 24. This is to ensure that the previous integrity checksum to be used later does not change once it has been computed.

[0026] Figure 3 illustrates a block diagram of one embodiment according to the invention. In Figure 3 all components have been disclosed separately, but it is obvious to a person skilled in the art that components may be implemented also in the form of a computer program. The system functions according to the method presented in Figure 2. Thus, the functionality is not described in detail.

[0027] The system according to the invention comprises a data source 30, a signing entity 31, a database 32, a database administration console 33 and a verification entity 34. Data source 30 may be any information system that produces data that needs to be stored on database 32. Signing entity 31 is for example a computer program running in a computer that is connected to database system 32 or a program module in database system 32. Database 32 and database admini-

stration console 33 may be any general-purpose database system, such as the Oracle database system. Verification entity 34 is similar to signing entity 31. If public key infrastructure is used, signing entity 31 has the secret key and verification entity 34 has the corresponding public key

[0028] It is obvious to a person skilled in the art that with the advancement of technology, the basic idea of the invention may be implemented in various ways. The invention and its embodiments are thus not limited to the examples described above; instead they may vary within the scope of the claims.